

Bài tập tuần 02

Vòng đời phần mềm & Lập trình với cơ sở dữ liệu

Mục tiêu

- Thực hiện các bài tập (câu hỏi) về nội dung Vòng đời phần mềm
- Xác định các đầu vào (input) và kết quả (output) cho nhóm yêu cầu nghiệp vụ của bài toán (case study)
- Lập trình với cơ sở dữ liệu:
 - o Xây dựng ứng dụng Java trên công cụ Netbeans
 - o Thiết kế, xây dựng cơ sở dữ liệu với phpMyAdmin trên MySQL
 - o Lập trình ứng dụng Java kết nối MySQL, thực thi các truy vấn SQL

Đánh giá

- Hoàn thành các bài tập về nội dung Vòng đời phần mềm, nắm được đặc điểm chính của các mô hình phát triển phần mềm khác nhau.
- Các nhóm sinh viên xác định được thông tin cơ bản (input / output) cho nghiệp vụ mà nhóm đã lựa chọn để phát triển
- Hoàn thành bài thực hành lập trình với cơ sở dữ liệu

Phần I:

Bài 1.1

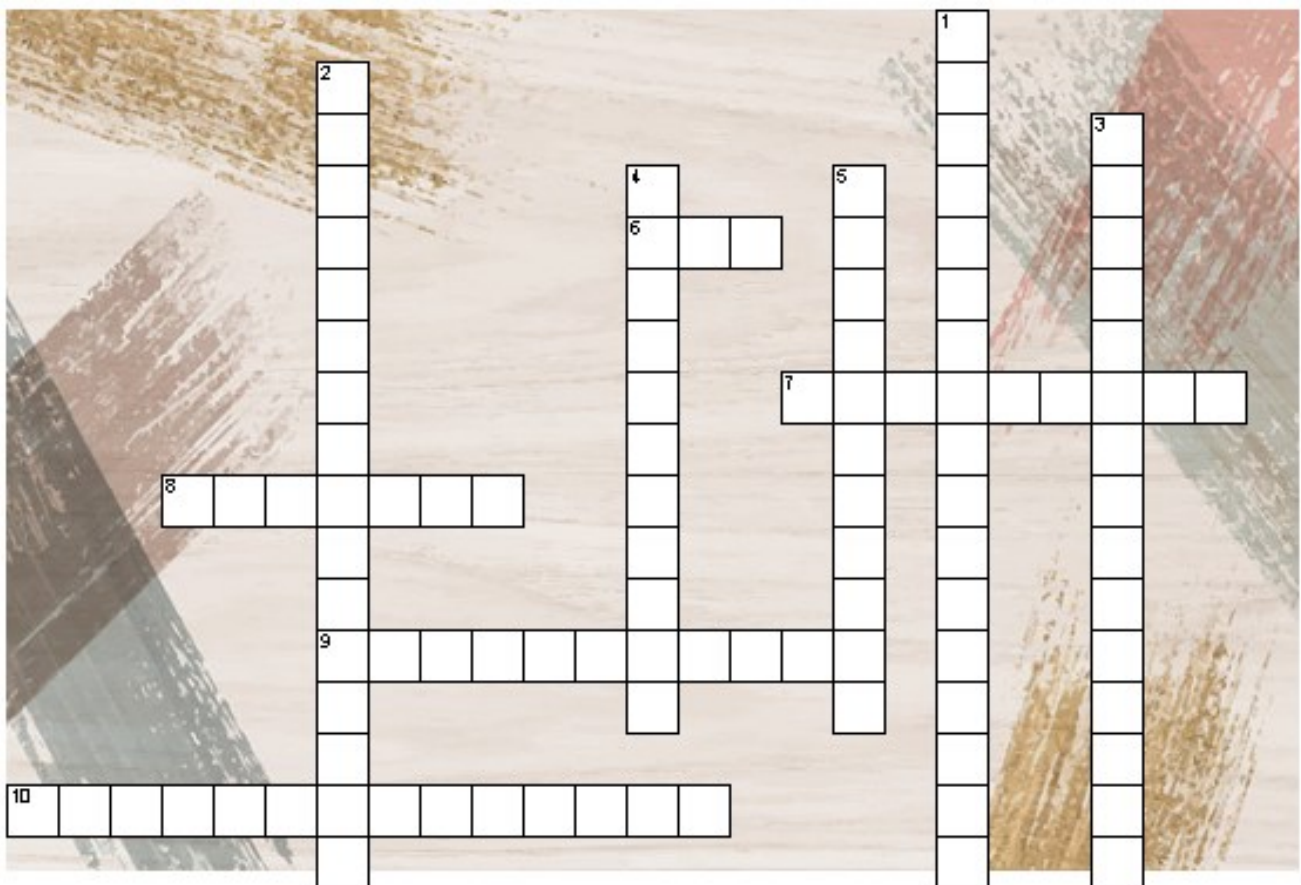
- a) Mô hình bản mẫu (prototyping model) của phát triển phần mềm là ...
 1. Một cách tiếp cận hợp lý khi yêu cầu được định nghĩa rõ ràng
 2. Một cách tiếp cận hữu ích khi khách hàng không thể định nghĩa yêu cầu rõ ràng
 3. Cách tiếp cận tốt nhất cho những dự án có đội phát triển lớn
 4. Tất cả các phương án trên đều sai
- b) Bước đầu tiên trong vòng đời phát triển phần mềm (Software Development Life Cycle) là?
 1. Xác định các nhu cầu và ràng buộc
 2. Viết phần mềm
 3. Vận hành hệ thống để loại bỏ các khiếm khuyết
 4. Nâng cao sản phẩm sau khi đã triển khai

c) Tình huống nào không phù hợp để có thể áp dụng mô hình thác nước? (chọn nhiều)

1. Khó khăn trong việc bổ sung yêu cầu mới trong các pha sau của tiến trình
2. Các yêu cầu được xác định rõ ràng, đầy đủ ngay từ đầu
3. Khách hàng muốn có sản phẩm vận hành được sớm
4. Khó thu thập đầy đủ yêu cầu ngay ở giai đoạn đầu của dự án

Bài 1.2

Hãy giải ô chữ dưới đây với các gợi ý kèm theo?



Các gợi ý cho ô chữ:

Across

6. Mô hình phát triển tăng dần từng bước, chia thành các team thực hiện đầy đủ các pha. Mỗi chu trình phát triển rất ngắn từ 60-90 ngày
7. Các yêu cầu có tính ổn định cao. Các pha của tiến trình phát triển thực hiện tuần tự từng bước, không quay trở lại pha trước đó
8. Giai đoạn kiểm tra logic bên trong và chức năng bên ngoài để đảm bảo kết quả đầu ra chính xác
9. Giai đoạn đáp ứng những thay đổi, nâng cấp phần mềm đã phát triển do sự thay đổi của môi trường, nhu cầu
10. Có ưu điểm là khả năng tái sử dụng phần mềm thông qua các hoạt động: phân tích yêu cầu, thiết kế với sử dụng lại, phát triển và tích hợp, thẩm định

Down

1. Thời kỳ tính từ khi phần mềm được sinh (tạo) ra cho đến khi chết đi
2. Sự kết hợp mô hình tuần tự và ý tưởng lặp lại của chế bản mẫu
3. Tập hợp các hành động, phương thức, thực hành, thay đổi mà người ta dùng để duy trì và phát triển phần mềm và các thành phần liên quan
4. Mô hình có thể sử dụng như "hệ sơ khai" để thu thập yêu cầu người dùng qua các thiết kế nhanh
5. Rủi ro được xem xét kỹ ở mỗi lần lặp tiến trình

Bài 1.3

Hãy so sánh các mô hình phát triển phần mềm: Thác nước, Chế thử, Gia tăng, Xoắn ốc, Phát triển dựa trên thành phần

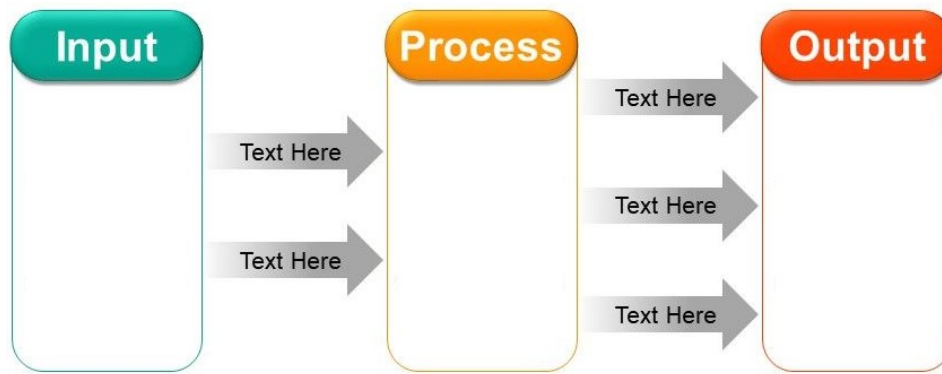
	Đặc điểm chính	Ưu điểm	Nhược điểm	Tình huống áp dụng phù hợp
Mô hình Thác nước				
Mô hình Chế thử				
Mô hình Gia tăng				
Mô hình Xoắn ốc				
Mô hình dựa trên thành phần				

Phần II: Phân tích thông tin cơ bản (input / output) cho nghiệp vụ bài toán (case study)

Bài 1.4

- Các nhóm thảo luận để xác định đầu vào (input) và kết quả (output) cho nhóm yêu cầu nghiệp vụ của bài toán (case study) mà nhóm đã lựa chọn.
- Phân rã mô tả của nhóm yêu cầu nghiệp vụ thành các nghiệp vụ nhỏ hơn.
- Với mỗi nghiệp vụ nhỏ này thực hiện phân tích các thông tin cụ thể

- Có thể vẽ sơ đồ thể hiện kết quả phân tích:

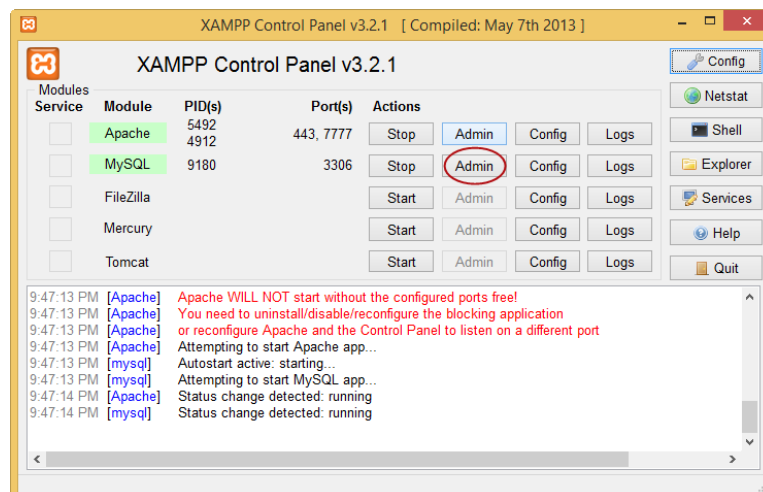


Hoặc thể hiện dưới dạng bảng:

Input	Process	Output

Phần III: Thực hành lập trình với cơ sở dữ liệu

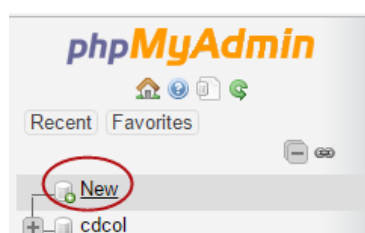
Bước 1: Khởi động **XAMPP Control Panel** → **Start** các dịch vụ Apache và MySQL → chọn **Admin** để mở công cụ quản trị phpMyAdmin cho cơ sở dữ liệu MySQL



- Hoặc có thể mở trình duyệt và truy cập vào: <http://localhost/phpmyadmin>

Bước 2: Tạo cơ sở dữ liệu

- Tại trang quản trị phpMyAdmin → chọn **New**



- Nhập tên database và chọn Create (nhớ chọn Collation là **utf8_unicode_ci**)

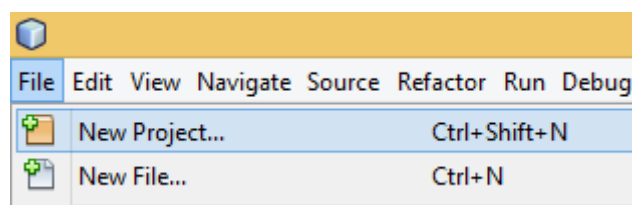


- Tạo một bảng dữ liệu và nhập các dữ liệu mẫu vào bảng

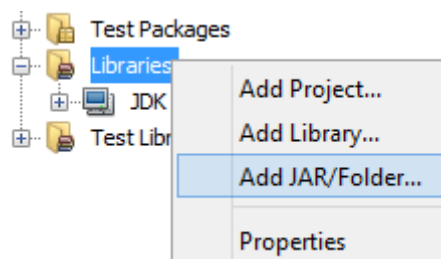
Name	Type	Collation	Attributes	Null	Default	Comments	Extra
ID	int(11)			No	None		AUTO_INCREMENT
userName	varchar(100)	utf8_unicode_ci		Yes	NULL		
passwd	varchar(100)	utf8_unicode_ci		Yes	NULL		

Bước 3: Xây dựng ứng dụng Java kết nối với MySQL

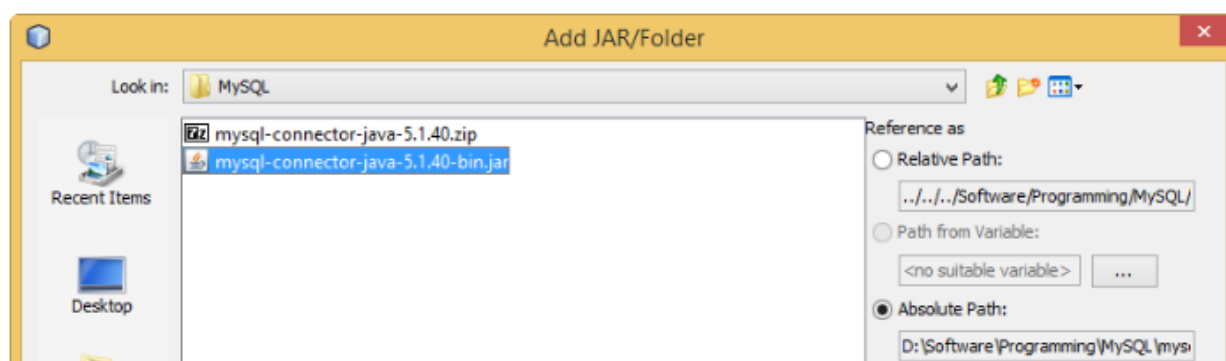
- Khởi động Netbeans -> **File -> New Project ...**



- Tải **JDBC Driver cho MySQL** và thêm vào project
(<https://dev.mysql.com/downloads/file/?id=465644>)



- Chọn tập tin jar -> chọn **Open**



- Viết code để truy vấn dữ liệu từ bảng trong cơ sở dữ liệu và hiển thị ra console (chú ý import các package cần thiết và sử dụng try/catch để bắt ngoại lệ khi thực thi các thực hiện các thao tác với CSDL / truy vấn SQL)

```

-Connection conn = null;
-Statement st = null;
-ResultSet rs = null;
-try {
-    String dbURL = "jdbc:mysql://localhost/tên_database";
-    String username = "root";
-    String password = "";
-    //Tạo kết nối
-    conn = DriverManager.getConnection(dbURL, username, password);
-    if (conn != null) {
-        System.out.println("Kết nối thành công");
-    }

-    // Câu lệnh xem dữ liệu
-    String sql = "select * from tên_bảng";
-    // Tạo đối tượng thực thi câu lệnh Select
-    st = conn.createStatement();
-    // Thực thi
-    rs = st.executeQuery(sql);

-    // Nếu không có dữ liệu trong bảng
-    if (rs.isBeforeFirst() == false) {
-        JOptionPane.showMessageDialog(this, "Bảng không có dữ liệu!");
-        return;
-    }
-    // Xử lý dữ liệu
-    while (rs.next()) {
-        //Sử dụng phương thức rs.getXX("Tên cột trong bảng")
-    }
-} catch (SQLException e) {
-    e.printStackTrace();
-}

```

Nội dung bài tập tự làm

- Thực hành xây dựng chương trình java với các lệnh SQL cơ bản (SELECT, INSERT, UPDATE, DELETE)

HẾT